



PROJEKT ZALICZENIOWY

JAVA - Programowanie Sieciowe



GRUPA L5

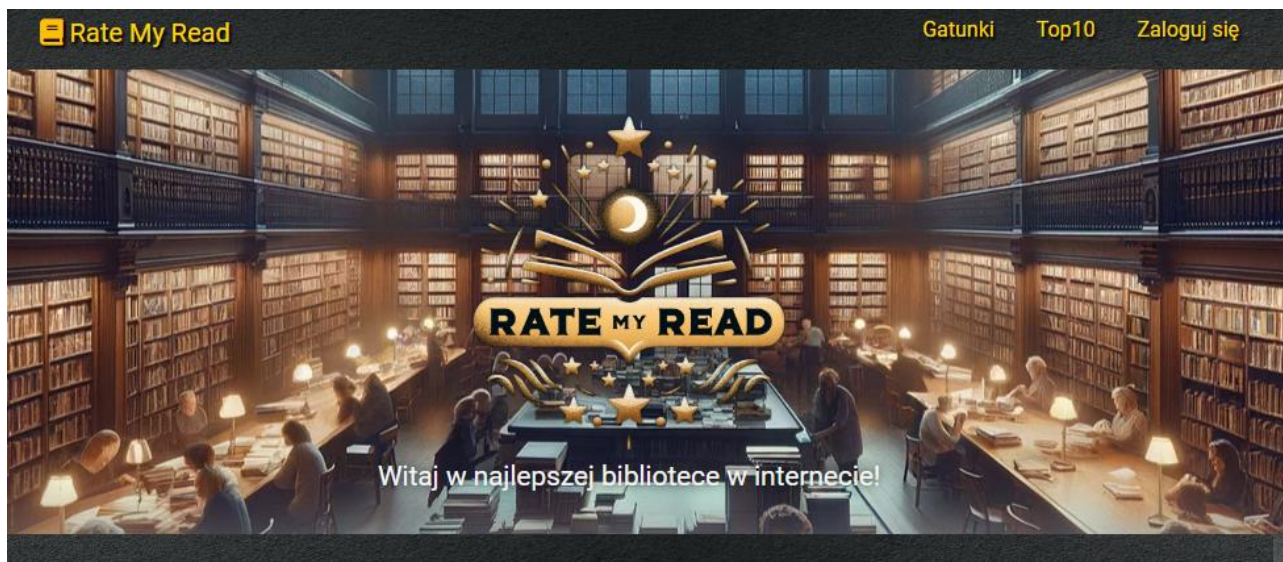
PIOTR HADAŁA
ARKADIUSZ NOWAK
PAWEŁ JĘDRZEJCZAK
NATALIA GÓROWSKA

Spis treści

1. Wstęp	2
2. Użyte technologie	2
3. Uruchomienie projektu	2
4. Token.....	3
Adres do Tokena	3
5. BookApiController.....	3
Adres do API.....	5
Otrzymanie książki po ID	6
Książki rekomendowane	6
Dodawanie książek	7
6. GenreApiController	8
Adres do API.....	9
Otrzymanie informacji o gatunku po jego nazwie	10
Dodawanie informacji o gatunku książki.....	10
7. Walidacja zapytań	11
8. Baza danych	12
9. Podział na mikroserwisy	14
Zawartość Rest-serivce:	14
BookService.java	14
<i>findBookById(long id)</i>	14
<i>findBooksByGenreName(String genre)</i>	14
<i>findAllBooks()</i>	14
<i>findTopBooks()</i>	14
<i>addBook(BookSaveDto bookToSave)</i>	15
<i>createBook(BookSaveApiDto bookToSave)</i>	15
GenreService.java	15
<i>FindGenreByName(String name)</i>	15
<i>FindAllGenres()</i>	15
<i>addGenre(GenreDto genre)</i>	15
RatingService.java	15
UserService.java	16
Funcjonalności Web-Service.....	17
Ocenianie książek	21

1. Wstęp

Projekt robiony na ocenę 5.0. W zadaniach na ocenę 3.0 wykonano polecenia z wariantu numer 1. W tym pliku opisany znajduje się dokumentacja całego projektu. Są tu utworzone dwa API, jedno do zwracania danych na temat książek znajdujących się w bazie danych, a drugie natomiast zwraca informacje odnośnie gatunków książek.



2. Użyte technologie

Apache Maven (wersja 4.0.0)

<https://maven.apache.org/download.cgi>

Spring Boot (wersja 3.2.1)

<https://start.spring.io>

3. Uruchomienie projektu

Na samym początku musimy stworzyć pustą bazę w mysql, o nazwie **springlibrary**(można też nadać jej inną nazwę, ale wtedy trzeba ją też wpisać w pliku application.properties w folderze rest-service i web-service).

W katalogu głównym projektu widzimy cztery foldery: eureka-service, gateway-service, rest-service i web-service. Każdy z nich otwieramy jako oddzielny projekt, najlepiej w IntelliJ Idea. Uruchamiamy wszystkie service i możemy już wejść na naszą stronę, domyślnie wpisując w przeglądarkę `http://localhost:8080`

4. Token

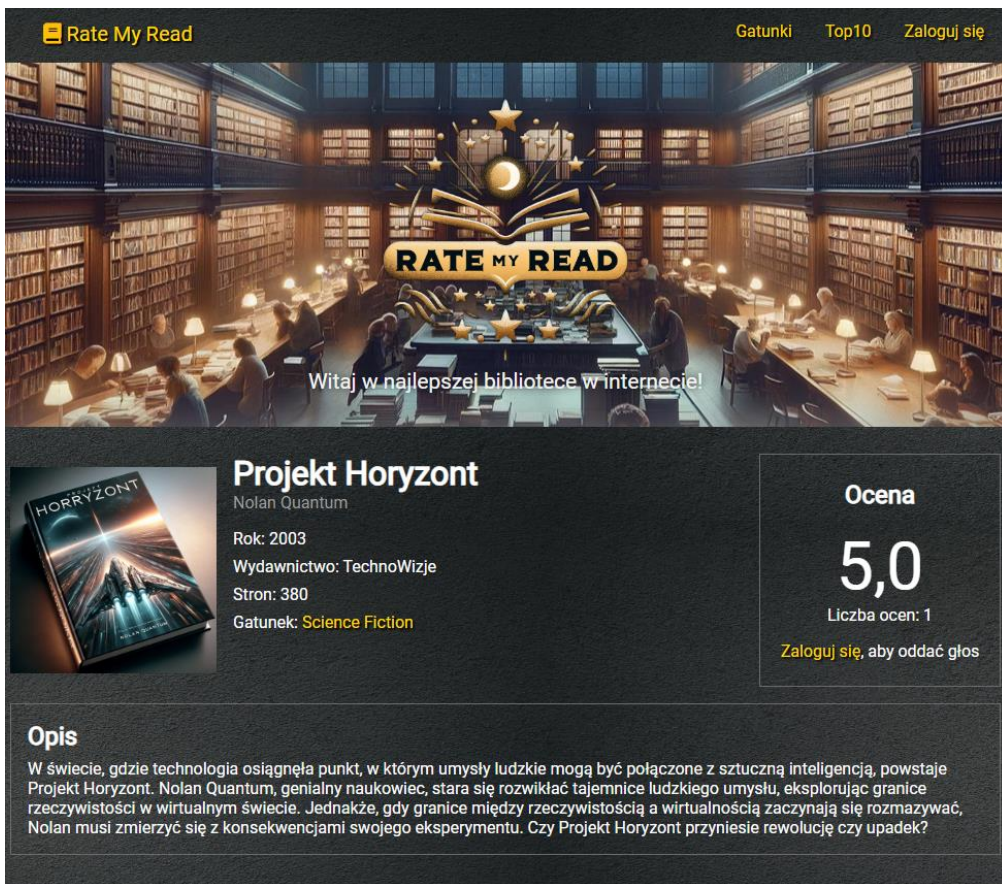
Adres do Tokena

GET /api/auth

- **Parametry:** UserDto – obiekt reprezentujący użytkownika(username, password)
- **Zwracane wartości:** token



5. BookApiController



Klasa BookApiController to kontroler, obsługujący zapytania dotyczące książek w API biblioteki.

Adnotacja @RestController określa, że jest to kontroler REST, który obsługuje zapytania HTTP.

Adnotacja @RequestMapping określa bazową ścieżkę dla wszystkich endpointów definiowanych w tej klasie.

Poniżej znajduje się opis poszczególnych endpointów tego kontrolera.

Plik BookApiController.java:

```
package pl.library.libraryonlineapiservice.web.api;

import org.springframework.http.HttpStatus;
import org.springframework.http.MediaType;
import org.springframework.http.ResponseEntity;
import org.springframework.web.bind.annotation.*;
import org.springframework.web.server.ResponseStatusException;
import org.springframework.web.servlet.support.ServletUriComponentsBuilder;
import pl.library.libraryonlineapiservice.domain.book.BookService;
import pl.library.libraryonlineapiservice.domain.book.dto.BookDto;
import pl.library.libraryonlineapiservice.domain.book.dto.BookSaveApiDto;

import java.net.URI;
import java.util.List;

@RestController
@RequestMapping(
    path = "/api/books",
    produces = {
        MediaType.APPLICATION_JSON_VALUE,
        MediaType.APPLICATION_XML_VALUE
    })
public class BookApiController {
    private final BookService bookService;

    public BookApiController(BookService bookService) {
        this.bookService = bookService;
    }

    @GetMapping()
    public List<BookDto> getAllBooks() {
        return bookService.findAllBooks();
    }

    @GetMapping("/{id}")
    public BookDto getSingleBook(@PathVariable long id) {
        BookDto book = bookService.findBookById(id)
            .orElseThrow(() -> new ResponseStatusExcep-
tion(HttpStatus.NOT_FOUND));
        return book;
    }

    @GetMapping("/recommended")
    public List<BookDto> getRecommendedBooks() {
        return bookService.findAllPromotedBooks();
    }

    @PostMapping
    ResponseEntity<BookDto> addBook(@RequestBody BookSaveApiDto bookDto) {
        BookDto addedBook = bookService.createBook(bookDto);
        URI savedJobOfferUri = ServletUriComponentsBuilder.fromCurrentRequest()
            .path("/{id}")
    }
```



```

        .buildAndExpand(addedBook.getId())
        .toUri();
    return ResponseEntity.created(savedJobOfferUri).body(addedBook);
}
}

```

Adres do API

GET /api/books

Aby otrzymać dane z API należy dodać do adresu **/api/books**. Otrzymamy wtedy listę wszystkich pozycji znajdujących się w bazie.

```

29 @GetMapping()
30 public List<BookDto> getAllBooks() { return bookService.findAllBooks(); }

```

GET localhost:8080/api/books

Params Authorization Headers (8) Body ● Pre-request Script

Body Cookies Headers (12) Test Results

Pretty Raw Preview Visualize JSON

```

1 [
2   {
3     "id": 1,
4     "title": "Kwazar: Sekret Gwiazdy Śmierci",
5     "author": "Olivia Lark",
6     "publisher": "Kosmiczne Opowieści",
7     "release_year": 2003,
8     "pages": 384,
9     "description": "W odległej przyszłości, ludzkość
10      w pobliżu najjaśniejszego kwazara w galaktyce
11      kosmiczna, zostaje wysłana na misję badawczą
12      "img": "Kwazar.png",
13      "genre": "Fantasy",
14      "promoted": false,
15      "avgRating": 3.0,
16      "ratingCount": 1
17   },
18   {
19     "id": 2,
20     "title": "Tajemnica Królestwa Wiatrów",
21     "author": "Marcus Zephyr",
22     "publisher": "Zakłete Strony",
23     "release_year": 2003,
24     "pages": 432,
25     "description": "W magicznym świecie pełnym zaklę
26      niezwyklej mocy. Od wieków, Królestwo to by
27      rozprzestrzeniać na kontynencie, wiedźma o i
28      "img": "Tajemnica.png",
29      "genre": "Kryminał",
30      "promoted": false,
31      "avgRating": 4.0,
32      "ratingCount": 1
33   },
34 ]

```

Otrzymanie książki po ID

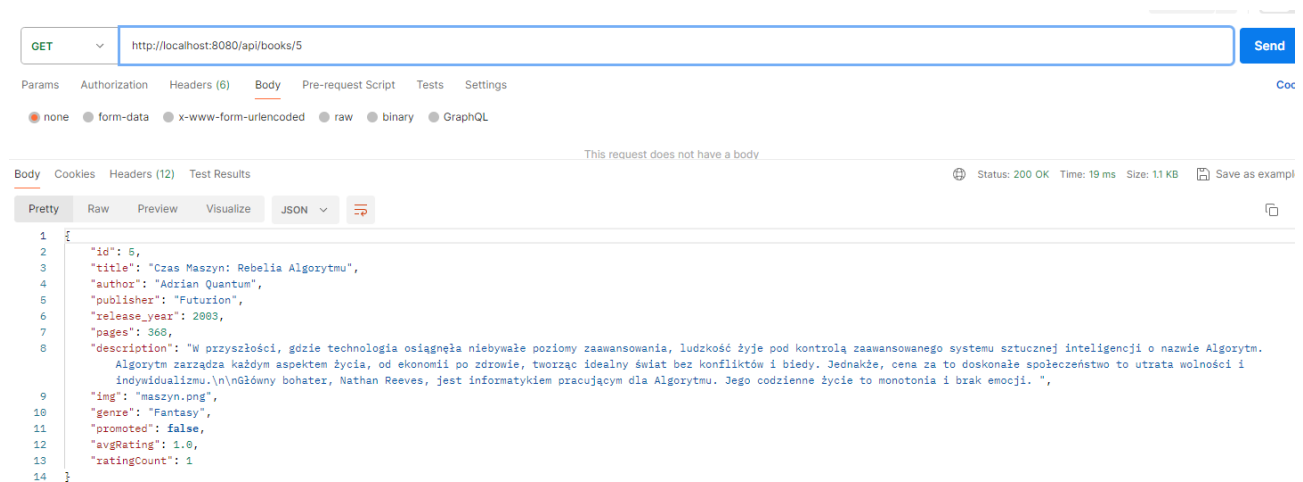
GET /api/books/{id}

Po wpisaniu /api/books/{id}, np. /api/books/5 otrzymamy informacje o pozycji z ID równym 5.

```
34 @GetMapping("/{id}")
35 public BookDto getSingleBook(@PathVariable long id) {
36     BookDto book = bookService.findBookById(id)
37         .orElseThrow(() -> new ResponseStatusException(HttpStatus.NOT_FOUND));
38     return book;
39 }
```

Zwraca pojedynczą książkę o podanym id. Jeśli jej nie znaleziono, zwraca błąd 404 (NOT_FOUND).

- **Parametry:** id – identyfikator książki
- **Zwracane wartości:** obiekt book, reprezentujący znaną książkę



Książki rekomendowane

GET /api/books/recommended

Ścieżka /api/books/recommended zwróci nam pozycje, dla których wartość recommended wynosi true.

```
40 @GetMapping("/recommended")
41 public List<BookDto> getRecommendedBooks() {
42     return bookService.findAllPromotedBooks();
43 }
```

```
← localhost:8080/api/books/recommended

This XML file does not appear to have any style information associated with it. The document tree is shown below:

<?xml version="1.0"?>
<list>
  <item>
    <id>11</id>
    <title>Układ Słoneczny: Nowa Granica</title>
    <author>Serena Cosmos</author>
    <publisher>AstrOpoWiedzi</publisher>
    <release_year>2016</release_year>
    <pages>420</pages>
    <description>W czasach, gdy kolonie ludzkie rozkwitają w różnych częściach Układu Słonecznego, Serena Cosmos, pilotka statku kosmicznego, staje w obliczu tajemniczej anomalii, która zagraża stabilności całego systemu. Razem z ekipą musi odkryć źródło tej anomalii i zapobiec katastrofie. Czy nowa granica przyniesie nadzieję czy zagładę?</description>
    <img>Układ_Słoneczny.png</img>
    <genre>Science Fiction</genre>
    <promoted>true</promoted>
  </item>
  <item>
    <id>19</id>
    <title>Dziedzictwo Kamienia</title>
    <author>Aria Stonemage</author>
    <publisher>Kamienne Opowieści</publisher>
    <release_year>2003</release_year>
    <pages>400</pages>
    <description>W krainie, gdzie magia płynie z kamieni, Aria Stonemage odkrywa, że posiada zdolności do kontrolowania magii kamieni, co uznawane jest za dar od pradawnych bogów. Jej zadaniem jest ocalić świat przed inwazją potężnych istot z innej rzeczywistości, które chcą pochłonąć magię kamieni. Czy Aria zdoła zrealizować swoje przeznaczenie i ochronić dziedzictwo kamienia?</description>
    <img>Dziedzictwo_Kamienia.png</img>
    <genre>Fantasy</genre>
    <promoted>true</promoted>
  </item>
  <item>
    <id>26</id>
    <title>Zakazane Miasto</title>
    <author>Nathan Blackwell</author>
    <publisher>Rocznice Katy</publisher>
    <release_year>2003</release_year>
    <pages>400</pages>
    <description>W tajemniczym zakazanym mieście, gdzie prawo nie obowiązuje, a korupcja sięga szczytu, detektyw Nathan Blackwell zostaje wysłany, aby zapanować nad rosnącym chaosem. W miarę jak grzechy przeszłości wypływają na powierzchnię, Blackwell musi stawić czoła moralnym dylematom i skomplikowanym relacjom, aby przywrócić sprawiedliwość w zakazanym mieście.</description>
    <img>Zakazane_Miasto.png</img>
    <genre>Kryminał</genre>
    <promoted>true</promoted>
  </item>
</list>
```

Dodawanie książek

POST /api/books

Dodaje nową książkę do bazy danych.

- Parametry: bookDto – obiekt, reprezentujący książkę
- Token
- Zwracane wartości: kod 201 (created), jeśli dodano książkę

POSTlocalhost:8080/api/booksSend

ParamsAuthorizationHeaders (9)BodyPre-request ScriptTestsSettingsCookiesBeautify

noneform-datax-www-form-urlencodedrawbinaryGraphQLJSON

```
1 {
2   "title": "Nowa książka",
3   "author": "Grupa L5",
4   "publisher": "UKEN",
5   "release_year": 2024,
6   "pages": 123,
7   "description": "W magicznym świecie pełnym smoków, elfów i zaklęć istnieje tajemnicza Smocza Pustynia, miejsce uważane za najniebezpieczniejsze i najbardziej niezbadane na kontynencie. Przez wieki, niewielu odważnych próbowało ją przemierzyć, ale nikt nie powrócił, by opowiedzieć o swoich przygodach. Aż do teraz.\n\nGłówną bohaterką opowieści jest Selena, młoda magini o niezwykłych zdolnościach, która marzy o odkryciu prawdy o Smoczej Pustyni.",
8   "img": "img.jpg",
9   "genre": "Fantasy",
10  "promoted": false
11 }
```

BodyCookiesHeaders (13)Test ResultsStatus: 201 CreatedTime: 25 msSize: 1.07 KBSave as example

PrettyRawPreviewVisualizeJSON

```
1 {
2   "id": 192,
3   "title": "Nowa książka",
4   "author": "Grupa L5",
5   "publisher": "UKEN",
6   "release_year": 2024,
7   "pages": 123,
8   "description": "W magicznym świecie pełnym smoków, elfów i zaklęć istnieje tajemnicza Smocza Pustynia, miejsce uważane za najniebezpieczniejsze i najbardziej niezbadane na kontynencie. Przez wieki, niewielu odważnych próbowało ją przemierzyć, ale nikt nie powrócił, by opowiedzieć o swoich przygodach. Aż do teraz.\n\nGłówną bohaterką opowieści jest Selena, młoda magini o niezwykłych zdolnościach, która marzy o odkryciu prawdy o Smoczej Pustyni.",
9   "img": "img.jpg",
10  "genre": "Fantasy",
11  "promoted": false,
12  "avgRating": 0.0,
13  "ratingCount": 0
14 }
```


6. GenreApiController



To API pozwoli nam na otrzymywanie informacji na gatunków książek które posiadamy w naszej bazie danych.

Plik GenreApiController.java:

```
package pl.library.libraryonlineapiservice.web.api;

import org.springframework.http.HttpStatus;
import org.springframework.http.MediaType;
import org.springframework.http.ResponseEntity;
import org.springframework.web.bind.annotation.*;
import org.springframework.web.server.ResponseStatusException;
import org.springframework.web.servlet.support.ServletUriComponentsBuilder;
import pl.library.libraryonlineapiservice.domain.genre.GenreService;
import pl.library.libraryonlineapiservice.domain.genre.dto.GenreDto;

import java.net.URI;
import java.util.List;

@RestController
@RequestMapping(
    path = "/api/genres",
    produces = {
        MediaType.APPLICATION_JSON_VALUE,
        MediaType.APPLICATION_XML_VALUE
    })
public class GenreApiController {
    private final GenreService genreService;

    public GenreApiController(GenreService genreService) {
        this.genreService = genreService;
    }

    @GetMapping("/{name}")
    public GenreDto getGenre(@PathVariable String name) {
        GenreDto genre = genreService.findGenreByName(name)
            .orElseThrow(() -> new ResponseStatusException(HttpStatus.NOT_FOUND));
        return genre;
    }

    @GetMapping
    public List<GenreDto> getGenreList() {
        return genreService.findAllGenres();
    }

    @PostMapping
```

```

    ResponseEntity<GenreDto> addGenre(@RequestBody GenreDto genreDto) {
        GenreDto addedGenre = genreService.addGenre(genreDto);
        URI savedJobOfferUri = ServletUriComponentsBuilder.fromCurrentRequest()
            .path("/{id}")
            .buildAndExpand(addedGenre.getId())
            .toUri();
        return ResponseEntity.created(savedJobOfferUri).body(addedGenre);
    }
}

```

Adres do API

GET /api/genres

Wyszukuje wszystkie gatunki i zwraca je w postaci listy.

- **Zwracane wartości:** obiekt List<Genre> będący listą wszystkich znalezionych gatunków

```

+ hadala-p +1
@RestController
@RequestMapping(
    path = @"/api/genres",
    produces = {
        MediaType.APPLICATION_JSON_VALUE,
        MediaType.APPLICATION_XML_VALUE
    })

```

```

37 @GetMapping()
38 public List<GenreDto> getGenreList() { return genreService.findAllGenres(); }
41 }

```

GET localhost:8080/api/genres

Params Authorization Headers (8) Body Pre-request Script Tests Settings

Body Cookies Headers (12) Test Results

Pretty Raw Preview Visualize JSON

```

1 [
2   {
3     "id": 1,
4     "name": "Science Fiction",
5     "description": "Gatunek łączący naukę i fikcję, często ekspl
        takich jak zaawansowane technologie, podróże międzyplane
6   },
7   {
8     "id": 2,
9     "name": "Fantasy",
10    "description": "Gatunek literacki, w którym magia i nadprzyr
        wróżki, czarownicy i bohaterowie o niezwykłych mocach."
11  },
12  {
13    "id": 3,
14    "name": "Kryminał",
15    "description": "Literatura skupiająca się na przestępstwach,
        detektywem jako głównym bohaterem."
16  },
17  {
18    "id": 4,
19    "name": "Romans",
20    "description": "Gatunek literacki skupiający się na relacjach
        związków międzyludzkich."
21  },
22  {
23    "id": 5,
24    "name": "Dla dzieci",
25    "description": "Literatura przeznaczona głównie dla dzieci,
        humorystyczne i ilustracje wspierające narrację."
26  },
27  {
28    "id": 6,
29    "name": "Nauka",

```

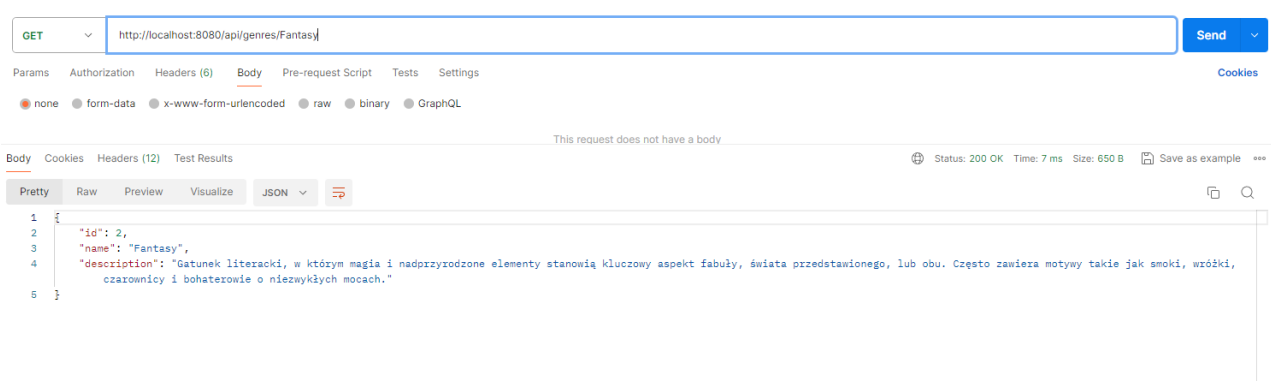
Otrzymanie informacji o gatunku po jego nazwie

GET /api/genres/{name}

Zwraca informacje o gatunku na podstawie jego nazwy.

- **Parametry:** name – nazwa gatunku
- **Zwracane wartości:** obiekt <GenreDto>, zawierający informacje o znalezionym gatunku

Po wpisaniu /api/genres/{name}, np. /api/genres/Fantasy otrzymamy informacje związane z rekordem o nazwie Fantasy.

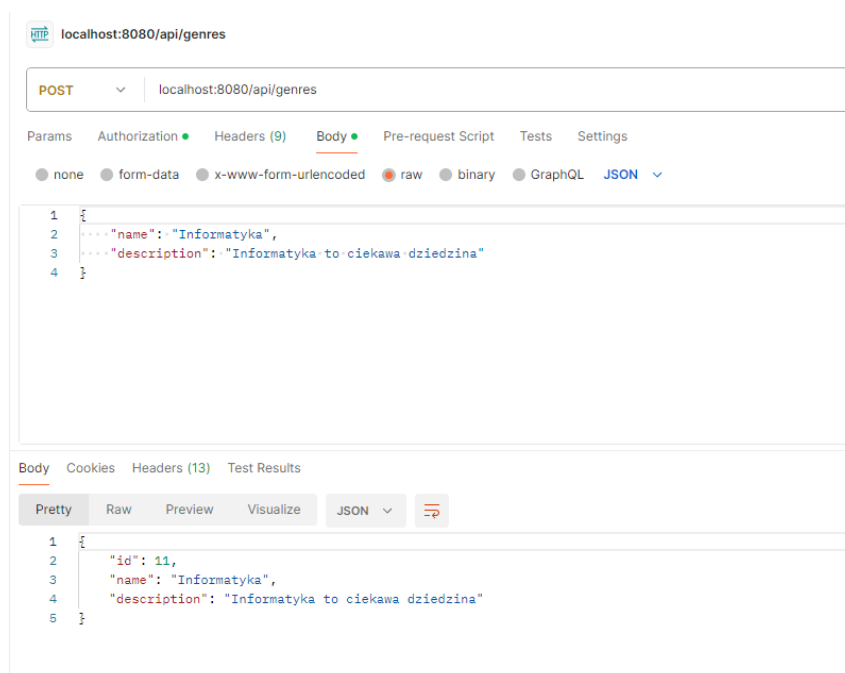


Dodawanie informacji o gatunku książki

POST /api/genres

Dodaje nowy gatunek do bazy danych.

- **Parametry:** GenreDto – obiekt, reprezentujący gatunek
- **Token**
- **Zwracane wartości:** kod 201 (created), jeśli dodano gatunek



7. Walidacja zapytań

Powyższe API zostały zabezpieczone przed nieprawidłowymi zapytaniami.

```
22 @GetMapping("/{id}")
23 public String getBook(@PathVariable long id, Model model) {
24     BookDto book = bookService.findBookById(id)
25         .orElseThrow(() -> new ResponseStatusException(HttpStatus.NOT_FOUND));
26     model.addAttribute("book", book);
27     return "book";
28 }
```

W przypadku gdy spróbujemy otrzymać dane o rekordzie, który nie istnieje, np. `/api/books/500` (szukanie książki o id 500 – nie istnieje taka w bazie), to otrzymamy komunikat, że nie istnieje taka strona.



8. Baza danych

master.xml

Ten plik XML używa narzędzia Liquibase do zarządzania zmianami w strukturze bazy danych. odnosi się do sytuacji, w jakiej wprowadzane są zmiany. W tym przypadku, zmiany danych testowych są stosowane tylko w środowisku oznaczonym jako "dev"- deweloperskie.

Pliki create[...].xml odpowiadają za zdefiniowanie struktury baz danych- określają jak mają być tworzone bazy.

Pliki [...].xml określają, jakie dane mają znajdować się w poszczególnych bazach.

001_create_table_genre.xml

Plik definiuje strukturę tabeli "genre" w bazie danych. Posiada ona następujące kolumny:

- id – ID gatunku książki
- name – nazwa gatunku, np. fantasy, romans
- description – opis gatunku

001_genres.xml

Plik definiuje dane początkowe, które będą znajdować się w tabeli "genre".

002_create_table_book.xml

Plik definiuje strukturę tabeli "books" w bazie danych. Posiada ona następujące kolumny:

- title – ID książki
- author – autor książki
- publisher – wydawnictwo, które opublikowało książkę
- release_year – rok wydania książki
- pages- ilość stron książki
- description – opis książki
- img – obraz, przedstawiający okładkę książki
- promoted- określa, czy książka jest promowana
- genre_id – ID gatunku książki

002_books.xml

Plik definiuje dane początkowe, które będą znajdować się w tabeli "books".

003_create_tables_users_and_roles.xml

Plik definiuje strukturę trzech powiązanych ze sobą tabel: "users", "user_role", oraz "user_roles" w bazie danych. Tabele te posiadają następujące kolumny:

- **users**
 - id – ID użytkownika
 - email – adres email użytkownika
 - password- hasło użytkownika
- **user_role**
 - id – ID roli użytkownika
 - name – nazwa roli
 - description – opis roli
- **user_roles**
 - user_id – ID użytkownika
 - role_id – ID roli

003_users_and_roles.xml

Plik definiuje dane początkowe, które będą znajdować się w tabelach “users”, “user_role” i “user_roles”.

004_create_table_book_ratings.xml

Plik definiuje strukturę tabeli “book_rating” w bazie danych. Posiada ona następujące kolumny:

- id – ID oceny
- user_id – ID użytkownika, który ocenia książkę
- book_id – ID ocenianej książki
- rating- wartość liczbową oceny

004_book_ratings.xml

Plik definiuje dane początkowe, które będą znajdować się w tabeli “book_rating”.

9. Podział na mikroserwisy

Aplikacja podzielona jest na mikroserwisy:

- eureka-service
- gateway-service
- rest-service
- web-service

Instances currently registered with Eureka			
Application	AMIs	Availability Zones	Status
GATEWAY-SERVICE	n/a (1)	(1)	UP (1) - localhost:GATEWAY-SERVICE:8080
REST-SERVICE	n/a (1)	(1)	UP (1) - localhost:REST-SERVICE:9000
WEB-SERVICE	n/a (1)	(1)	UP (1) - localhost:web-service:9001

Zawartość Rest-service:

BookService.java

Klasa odpowiada za zarządzanie książkami w systemie biblioteki. Poniżej znajduje się opis poszczególnych metod klasy.

findAllPromotedBooks()

Zwraca listę wszystkich książek, oznaczonych jako promowane.

- **Zwracane wartości:** Lista promowanych książek- obiekt <BookDto>

findBookById(long id)

Zwraca książkę z danym ID

- **Parametry:** id – ID książki
- **Zwracane wartości:** Obiekt Optional<BookDto>, zawierający znaną książkę.

findBooksByGenreName(String genre)

Zwraca listę książek należących do określonego gatunku.

- **Parametry:** genre – nazwa gatunku
- **Zwracane wartości:** Lista książek danego gatunku- obiekt List<BookDto>

findAllBooks()

Zwraca listę wszystkich książek dostępnych w systemie.

- **Zwracane wartości:** Lista wszystkich książek w formacie List<BookDto>

findTopBooks()

Zwraca listę książek z najwyższymi ocenami.

- **Parametry:** size- ilość książek do znalezienia
- **Zwracane wartości:** Lista znalezionych książek w formacie List<BookDto>

addBook(BookSaveDto bookToSave)

Dodaje nową książkę do bazy danych.

- **Parametry:** bookToSave – przechowuje wszystkie potrzebne dane o książce, którą chcemy dodać do systemu.

createBook(BookSaveApiDto bookToSave)

Dodaje nową książkę do bazy danych.

- **Parametry:** bookToSave – przechowuje wszystkie potrzebne dane o książce, którą chcemy dodać do systemu.
- **Zwracane wartości:** Obiekt <BookDto> reprezentujący nową książkę.

GenreService.java

Klasa odpowiada za zarządzanie gatunkami książek w systemie.

FindGenreByName(String name)

Pobiera informacje o gatunku na podstawie jego nazwy.

- **Parametry:** name – nazwa gatunku
- **Zwracane wartości:** Obiekt Optional<GenreDto> zawierający informację o znalezionym gatunku

FindAllGenres()

Zwraca listę wszystkich znalezionych gatunków.

- **Zwracane wartości:** List<GenreDto>- lista wszystkich gatunków

addGenre(GenreDto genre)

Dodaje nowy gatunek do bazy danych.

- **Parametry:** genre – informacje o nowym gatunku, który chcemy dodać
- **Zwracane wartości:** Obiekt GenreDto> zawierający informacje o dodanym gatunku

RatingService.java

Odpowiada za obsługę ocen książek, dodawanych przez użytkowników.

addOrUpdateRating(String userEmail, long bookId, int rating)

Dodaje lub aktualizuje ocenę, zamieszczoną przez użytkownika dla konkretnej książki.

- **Parametry:**
 - userEmail – adres email użytkownika
 - bookId – ID książki, którą ocenia użytkownik

- rating - wartość oceny, przyznawanej książce

getUserRatingForBook(String userEmail, long bookId)

Pobiera ocenę wystawioną przez użytkownika dla konkretnej książki.

- **Parametry:**
 - userEmail – adres email użytkownika
 - bookId – ID książki, którą ocenił użytkownik
- **Zwracane wartości:** Obiekt `Optional<Integer>` zawierający wartość oceny użytkownika dla danej książki

UserService.java

Klasa pozwala na zarządzanie użytkownikami biblioteki.

findCredentialsByEmail(String email)

Pobiera dane uwierzytelniające użytkownika na podstawie jego adresu email.

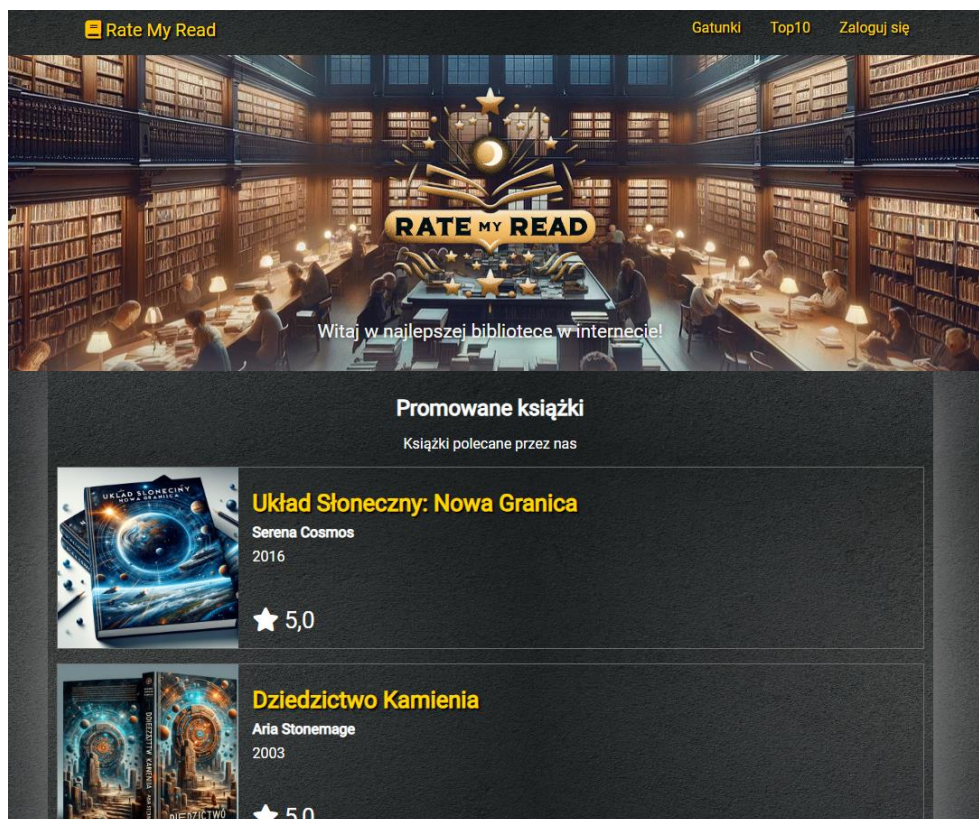
- **Parametry:** email – adres email użytkownika
- **Zwracane wartości:** email – Obiekt `Optional<UserCredentialsDto>` zawierający dane uwierzytelniające użytkownika

registerUserWithDefaultRole(UserRegistrationDto userRegistration)

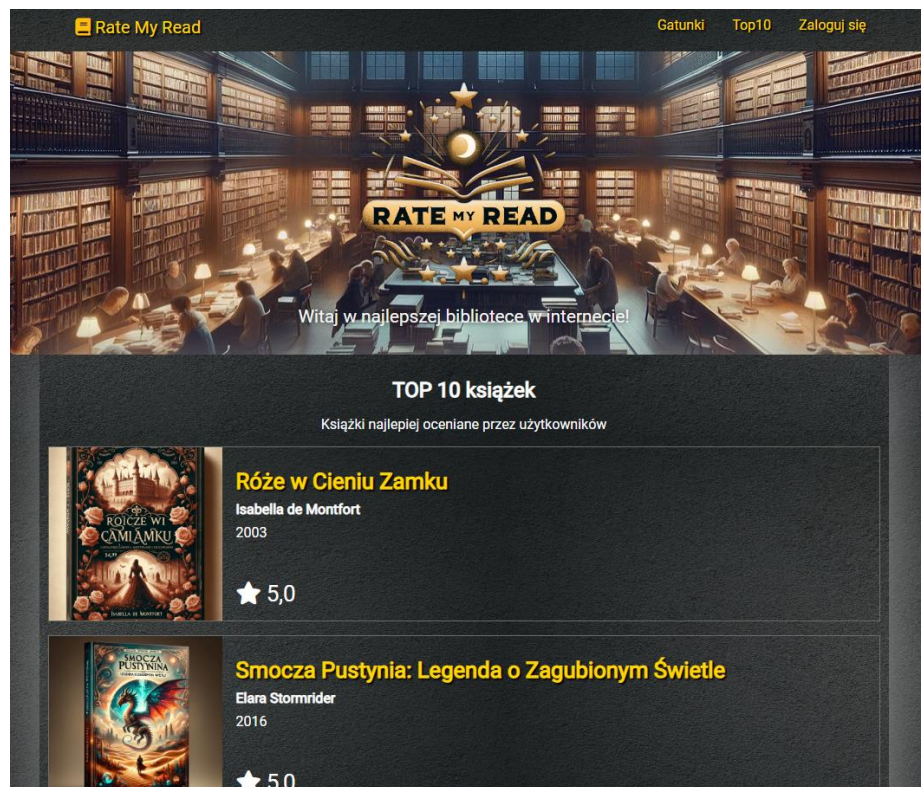
Rejestruje nowego użytkownika, przypisując mu domyślną rolę.

- **Parametry:** Obiekt `UserRegistrationDto` zawierający dane potrzebne do rejestracji nowego użytkownika

Funcjonalności Web-Service



Ranking książek



Gatunki książek

Poniżej znajdziesz listę gatunków książek w naszym katalogu. Wybierz coś dla siebie.

Science Fiction

Fantasy

Kryminał

Romans

Dla dzieci

Nauka

Przygodowe

Thriller

Horror

Biografia

Informatyka

Logowanie

 Rate My Read

Gatunki

Top10

Zaloguj się



Zaloguj się

Adres email

adres e-mail

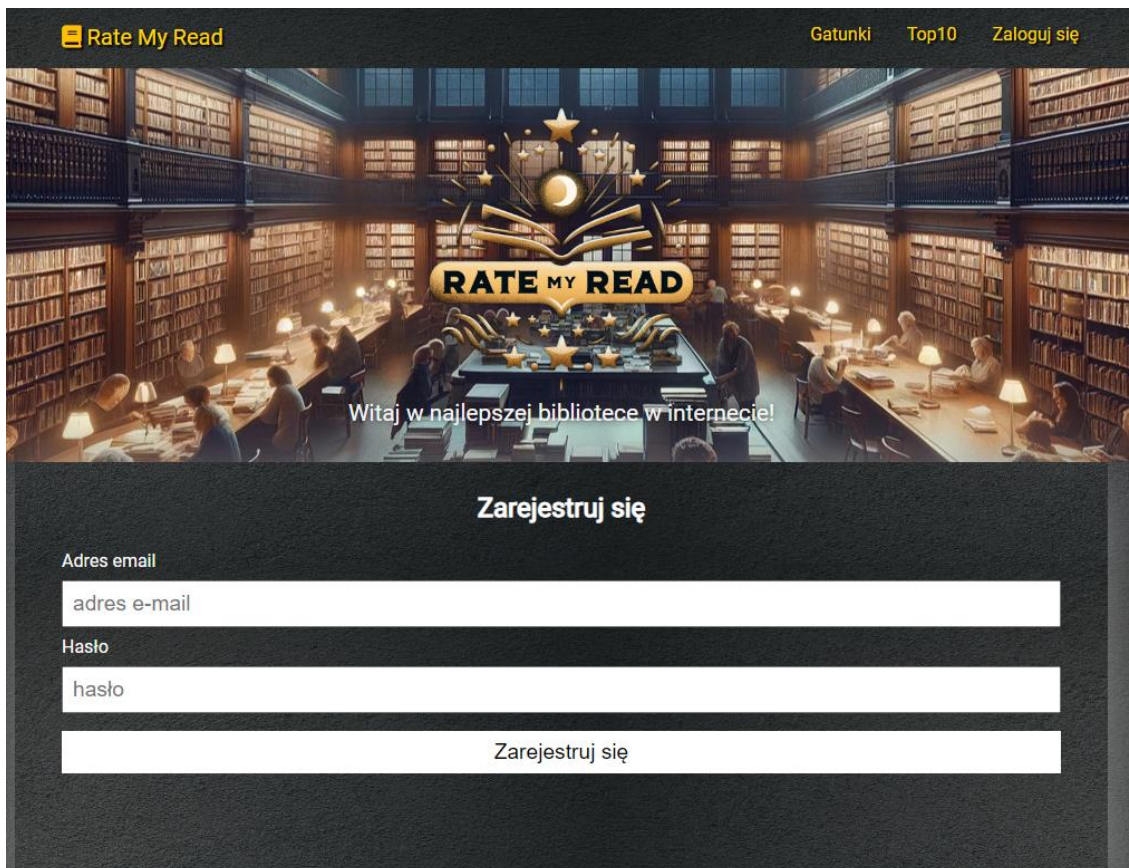
Hasło

hasło

Zaloguj się

Nie masz jeszcze konta? [Zarejestruj się](#)

Rejestracja



The registration form is overlaid on a background image of a large, multi-story library with high ceilings, wooden bookshelves, and people reading at tables. In the center of the library, there is a large, stylized logo of an open book with the text "RATE MY READ" and several stars above it. The text "Witaj w najlepszej bibliotece w internecie!" is written below the logo.

Rate My Read [Gatunki](#) [Top10](#) [Zaloguj się](#)

Witaj w najlepszej bibliotece w internecie!

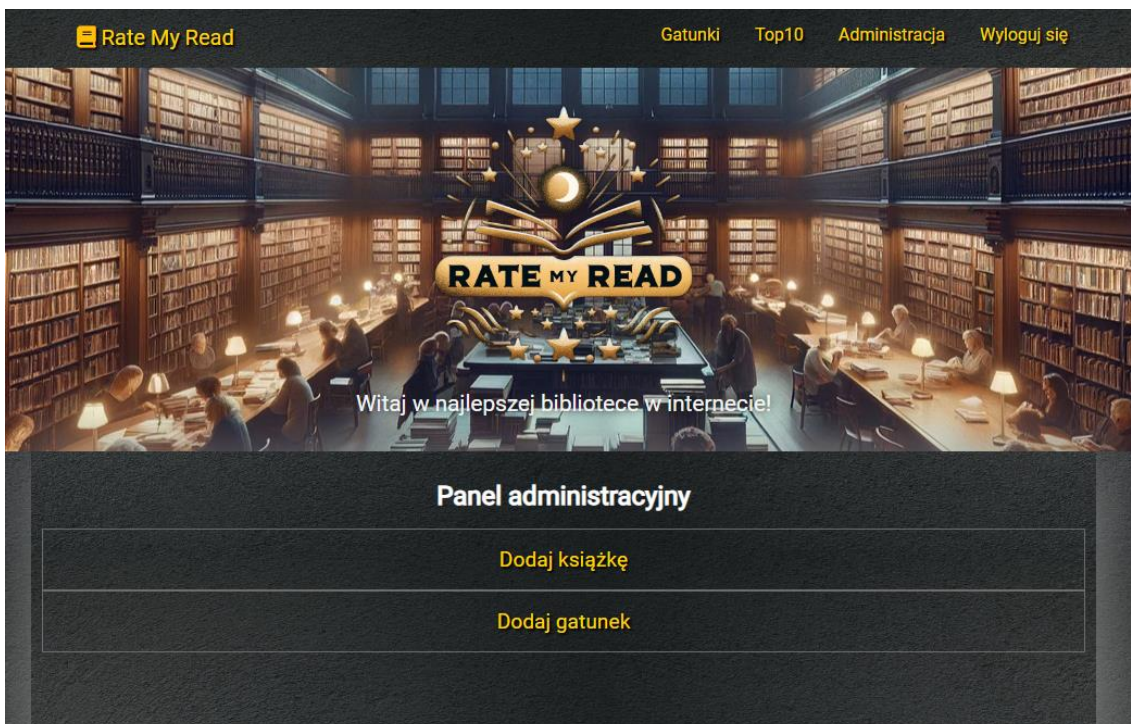
Zarejestruj się

Adres email

Hasło

Dodawanie książki lub gatunku(Admin lub Edytor)

*zdjęcie książki musi być w formie jpg, jpeg i być małym plikiem



The admin panel is overlaid on the same library background image as the registration form. It features a dark grey header with the "Rate My Read" logo and navigation links. The main content area is titled "Panel administracyjny" and contains two buttons: "Dodaj książkę" and "Dodaj gatunek".

Rate My Read [Gatunki](#) [Top10](#) [Administracja](#) [Wyloguj się](#)

Witaj w najlepszej bibliotece w internecie!

Panel administracyjny

[Dodaj książkę](#)

[Dodaj gatunek](#)

Dodaj nową książkę

Tytuł

Książki dla Psa

Autor

Monika Hanulak & Grażka Lange

Wydawnictwo

dwieSiostry

Rok premiery

2023

Ilość stron

89

Gatunek

Nauka

Promowanie na stronie głównej?

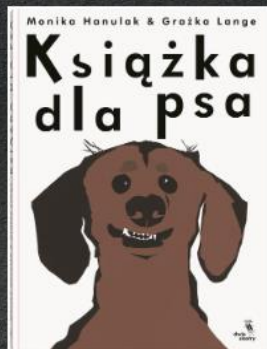


Obrazek

Wybierz plik książka.jpg

Pełny opis

Książki jest poradnikiem apropos zajmowanie się swoim psem|



Książki dla Psa

Monika Hanulak & Grażka Lange

Rok: 2023

Wydawnictwo: dwieSiostry

Stron: 89

Gatunek: Nauka

Ocena

Brak głosów

Twoja ocena



Opis

Książki jest poradnikiem apropos zajmowanie się swoim psem



Witaj w najlepszej bibliotece w internecie!

Dodaj nowy gatunek książki

Nazwa

Języki obce

Opis - max 500 znaków

W tej kategorii znajdziesz książki napisane w języku obcym

Dodaj

[Biografia](#)

[Informatyka](#)

[Języki obce](#)

Ocenianie książek



Zagubieni w Ciemności: Labirynt Grozy

Lost in Darkness

Rok: 2003

Wydawnictwo: Mroczne Przypadki

Stron: 310

Gatunek: **Horror**

Ocena

Brak głosów

Twoja ocena



Opis

Mroczne Przypadki przedstawiają Zagubionych w Ciemności, gdzie bohaterowie eksplorują tajemniczy labirynt, gdzie każdy zakręt może prowadzić do śmierci. Czy zdołają znaleźć wyjście z Labiryntu Grozy?